

# Agentes embebidos para la gestión de sonidos

Gabriel Villarrubia<sup>1</sup>, Alicia Fernández del Viso<sup>3</sup>, Carlos Rebate<sup>3</sup>, Javier Bajo<sup>2</sup>, Juan F. De Paz<sup>1</sup>, Carolina Zato<sup>1</sup>, Juan M. Corchado<sup>1</sup>

<sup>1</sup>Departamento de Informática y Automática, Universidad de Salamanca

Plaza de la Merced, s/n, 37008, Salamanca, España

{gvg, fcofds, carol\_zato, corchado}@usal.es

<sup>2</sup>Departamento de Inteligencia Artificial, Universidad Politécnica de Madrid

Campus Montegancedo, Boadilla del Monte, Madrid

{jbajo@fi.upm.es}

<sup>3</sup>Inclusion Unit-Indra,

Avda. Bruselas, 35 28108 Alcobendas. Madrid

{afernandezde, crebate}@indra.es

**Resumen.** La inteligencia ambiental ha presentado avances a lo largo de los últimos años. La incorporación de técnicas de inteligencia artificial ha posibilitado aspectos como el reconocimiento de patrones permitiendo así una mejor adaptación de estos sistemas. En este trabajo se presenta la arquitectura multiagente PANGEA que incorpora diferentes agentes y sensores encargados de dar soporte a usuarios en el hogar caso de determinadas alarmas o incidencias. El sistema incorpora agentes en dispositivos hardware arduino con módulos de reconocimiento y bandas luminosas.

**Palabras clave:** organizaciones virtuales, inteligencia ambiental, sensores.

## 1 Introducción

Durante los últimos años se han producido grandes avances en el campo de la Inteligencia Ambiental [2] [3], adquiriendo una importante relevancia en la vida cotidiana de las personas [6] [7] [8]. La inteligencia ambiental adapta la tecnología a las necesidades de la gente proponiendo 3 conceptos: computación ubicua, comunicación ubicua, interfaces de usuario inteligentes. Para alcanzar estos objetivos es necesario desarrollar nuevos frameworks y modelos que permitan acceder a la información con independencia de la localización. Las redes de sensores inalámbricas [4] [5][18] proporcionan una infraestructura capaz de distribuir las comunicaciones en entornos dinámicos incrementando la movilidad y la eficiencia con independencia de la localización. Las redes de sensores interconectan una gran cantidad de sensores y gestionan la información del entorno inteligente. En muchas ocasiones la gestión de la información se realiza de forma distribuida. Sin embargo, se hace necesario disponer

de sistemas distribuidos con capacidades suficientes para gestionar de forma eficiente las redes de sensores, y que dispongan de elementos software con cierto grado de inteligencia que puedan ser empotrados en los dispositivos y actuar tanto de forma autónoma como coordinada con el sistema distribuido. Los sistemas multi-agente son una alternativa muy adecuada para llevar a cabo este tipo de sistemas. Existen distintas propuesta que combinan los sistemas multi-agente y las redes de sensores para construir entornos inteligentes [11] [12] [13] [14][15] [16]. Sin embargo, no es posible encontrar una arquitectura multi-agente [17] existente que trabaje sobre el concepto de organizaciones virtuales [9] [23] y que proporcione agentes capaces de ser empotrados en los dispositivos.

En este artículo se presenta un modelo de agente capaz de ser ejecutado como software empotrado en un microcontrolador Arduino. El agente forma parte de la arquitectura PANGEA [22], una arquitectura multi-agente diseñada sobre la base de organizaciones virtuales, orientada a la creación de entornos inteligentes. PANGEA es una arquitectura multi-agente que proporciona varias herramientas para crear y manejar organizaciones virtuales de agentes. PANGEA incluye el Agente Sensor que puede ser empotrado en dispositivos con pocas capacidades computacionales permitiendo controlarlos. El agente Sensor se ha incluido en un dispositivo Arduino [10] facilitando así el manejo de la información obtenida de los diferentes dispositivos.

El artículo está estructurado de la siguiente forma: En primer lugar, se describe la arquitectura PANGEA, presentando en detalle la estructura del Agente Sensor. Tanto la arquitectura como el Agente Sensor son evaluados en un caso de estudio consistente en un entorno inteligente especialmente diseñado para el reconocimiento de sonidos de alarmas en interiores. Finalmente se presentan los resultados obtenidos en el caso de estudio y las conclusiones que se han podido extraer de esta investigación.

## **2 Plataforma PANGEA**

Con la evolución de los sistemas ubicuos y distribuidos, se hace necesario disponer de nuevas plataformas de agentes que faciliten el desarrollo de arquitecturas de agentes abiertas y capaces de desplegarse en cualquier dispositivo. PANGEA [22] es una plataforma de agentes que basándose en conceptos organizativos permite modelar e implementar toda clase de sistemas abiertos, fomentando la distribución de recursos y facilitando el control de todos los nodos donde los diferentes agentes son desplegados.

El desarrollo de los sistemas y tecnologías entorno a la Inteligencia Ambiental, entendidos como sistemas incluidos en extensos entornos con gran cantidad de elementos interactuando entre sí requiere de mecanismos necesarios para poder integrar los servicios, (autonomía, distribución, adaptabilidad, y capacidad de reacción) y que son también factores clave en los sistemas multi-agente. Por ello, se considera esencial disponer de mecanismos de control que permitan a los nuevos dispositivos incluirse en una plataforma única donde ser fácilmente integrados, gestionados y monitorizados. En este caso, PANGEA, gracias al modelo de agentes y organizaciones, ofrece las características necesarias para ejercer de plataforma base a la hora de desarrollar el sistema integral.

En PANGEA se dispone automáticamente de diversas organizaciones que son desplegadas automáticamente al iniciar el funcionamiento de la plataforma. Las organizaciones son las siguientes:

- Organización interfaz: en esta organización se despliegan todos aquellos agentes necesarios para la traducción o comunicación con agentes que no comparten el mismo ACL (*Agent Communication Language*) que los agentes propios de PANGEA. Los agentes PANGEA están implementados en Java y utilizan el estándar IRC (*Internet Relay Chat*) como lenguaje de comunicación. Actualmente, dentro de esta organización existen agentes con capacidades de traducción para agentes cuyos lenguajes de comunicación sean FIPA-ACL o KQML.
- Organización central de control: dentro de esta organización se incluyen los agentes que gestionan y controlan toda la plataforma, se verán en la siguiente sección.
- Organización de monitorización externa: en esta organización se incluyen aquellos agentes desplegados en dispositivos de monitorización externos pertenecientes al sistema de inteligencia ambiental como pueden ser módulos de reconocimiento de sonido. Se verá en detalle en las próximas secciones.

## 2.1 Agente Sensor

El objetivo principal de este apartado, es describir la funcionalidades que presenta el agente sensor que ha sido desarrollado, así como mencionar sus principales ventajas frente a los sistemas usados convencionalmente para desplegar agentes.

La utilización de los denominados sistemas embebidos, están diseñados para la realización de funciones muy específicas o sencillas, su principal uso está destinado a aplicaciones de cómputo en tiempo real. La principal característica que presenta la utilización de software empotrado frente a otros sistemas de computación, es el bajo coste en su proceso de fabricación, debido a que el procesador y memoria presentes en el dispositivo son de un tamaño reducido. Existen diferentes micro-controladores disponibles en el mercado, los más conocidos son Arduino [19], Propeller [20], Beagle Board [21] y Raspberry Pi [1].

La placa Arduino utilizada para el desarrollo del agente es el modelo Arduino Duemilanove [24] que se compone de los componentes mostrados en la figura 1:

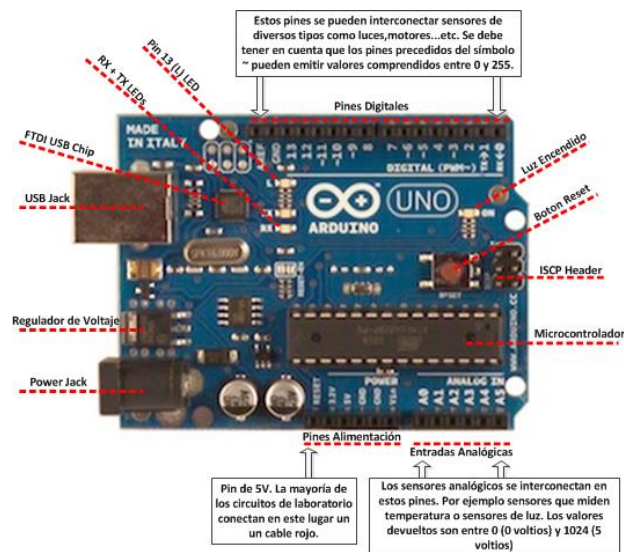


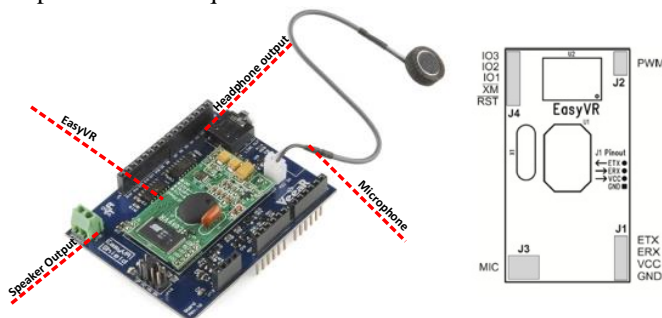
Fig. 14. Componentes de la placa Arduino

El modelo Duemilanove se encuentra caracterizado por la presencia del microcontrolador ATmega168 [25] o ATmega328. Presenta un total de 14 pines, de los cuales 6 pueden utilizarse como PWM (*Pulse Width Modulation*) [27], otros 6 pines pueden dedicarse para ámbitos analógicos. Además cuenta con un botón de reset, una conexión de tipo USB para interconectarse con un ordenador y dos pines para comunicarse con la placa mediante el protocolo ISCP [26].

### 2.1.2 Módulo Reconocimiento Voz

El tratamiento y posterior análisis acústico se realiza mediante una placa modelo EasyVR [28], que proporciona una conectividad muy sencilla con cualquier controlador mediante una comunicación de tipo UART [29]. Los sonidos a reconocer son almacenados en una memoria interna, con una capacidad de hasta 32 sonidos. Presenta un micrófono que permite captar de forma continua los sonidos del entorno y en consecuencia generar una respuesta.

La alimentación mínima necesaria para trabajar con este módulo es de 3,3V, siendo un buen candidato para trabajar conjuntamente con la placa Arduino Duemilanove. En la figura 2 se puede ver un esquema del módulo.



**Fig. 15.** Módulo de reconocimiento de voz para Arduino

### 2.1.3. Esquema Solución

El agente sensor presenta los siguientes componentes hardware que se pueden ver en la figura 3:

- Placa Arduino Duemilanove.
- Módulo de reconocimiento de voz EASYVR.
- Grupo de 100Leds de diferentes colores.

El micrófono se encuentra tomando muestras de los sonidos del entorno, cuando el patrón detectado, es similar al emitido por una sirena, el microcontrolador se encarga de activar y reproducir destellos luminosos requiriendo la atención del usuario mediante los led de alta visibilidad. Esto resulta útil para personas con discapacidad auditiva, inhabilitadas para escuchar una señal de alarma de un dispositivo acústico convencional.



**Fig. 16.** Integración de Arduino y módulo de reconocimiento de voz en PANGEA

Según se puede ver en la figura 3, el micro-controlador se encuentra conectado a una arquitectura PANGEA. Este sensor aprovecha la conectividad y los servicios que proporciona este sistema para el envío de alertas y notificaciones por diferentes mecanismos a los usuarios.

### 2.1.4. Algoritmos

En las siguientes imágenes se puede observar los algoritmos encargados del manejo de las advertencias luminosas, tanto para el encendido como para el apagado.

```

void TurnOnLED(){
    while (1){
        // Update the colors.
        byte time = millis() >> 2;
        for(byte i = 0; i < LED_COUNT; i++)
        {
            byte x = time - 8*i;
            colors[i] = (rgb_color){ x, 255 - x, x };
            //colors[i] = (rgb_color){ 0, 0, 0 };
        }
        // Write the colors to the LED strip.
        ledStrip.write(colors, LED_COUNT);
        delay(10);
    }
}

void TurnOffLED(){
    // Update the colors.
    byte time = millis() >> 2;
    for(byte i = 0; i < LED_COUNT; i++)
    {
        byte x = time - 8*i;
        //colors[i] = (rgb_color){ x, 255 - x, x };
        colors[i] = (rgb_color){ 0, 0, 0 };
    }
    // Write the colors to the LED strip.
    ledStrip.write(colors, LED_COUNT);
    delay(50);
}

```

El siguiente algoritmo se ocupa de comprobar si existen peticiones procedentes del servidor PANGEA hacia nuestro agente, posteriormente se procesa esa respuesta en función de la primitiva llegada.

```

void procesarSonido(){
  easyvr.setPinOutput(EasyVR::IO1, HIGH); // LED on (listening)
  read_from_server();
  easyvr.recognizeCommand(group);
  do
  {
    //WAIT
  }
  while (!easyvr.hasFinished());
  easyvr.setPinOutput(EasyVR::IO1, LOW); // LED off
  idx = easyvr.getWord();
  if (idx >= 0)
  {
    // built-in trigger (ROBOT)
    // group = GROUP_X; <-- jump to another group X
    return;
  }
  idx = easyvr.getCommand();
  if (idx >= 0)
  {
    // print debug message
    uint8_t train = 0;
    char name[32];
    Serial.print("Command: ");
    Serial.print(idx);
    if (easyvr.dumpCommand(group, idx, name, train))
    {
      Serial.print(" = ");
      Serial.println(name);
    }
    else
      Serial.println();
    easyvr.playSound(0, EasyVR::VOL_FULL);
    action();
  }
  else // errors or timeout
  {
    if (easyvr.isTimeout()){
      //read_from_server();
      Serial.println("Timed out, try again...");
    }
    int16_t err = easyvr.getError();
    if (err >= 0)
    {
      //Serial.print("Error ");
      //Serial.println(err, HEX);
    }
  }
}
}

void read_from_server(){
  String a_sentence;
  String response;
  String after_space;

  while (client.available())
  {
    c = client.read();
    a_sentence = "";
    //pull out characters from the client serial and assemble a string
    while (client.available() && c != '\n'){
      a_sentence = a_sentence + c;
      c = client.read();
    }
    response = command_response(a_sentence);
    //todo, stuff to detect if the message was sent from/to a pa or a channel
    // and direct the response back to that channel
    Serial.println(a_sentence);
    if (response != ""){
      printinBoth(response);
    }
  }
}
}

```

Este algoritmo es el más importante en el ámbito del reconocimiento de los patrones de sonido, su función principal es la de inicializar el módulo de reconocimiento EASYVR, una vez esperado el tiempo de inicialización necesario por el módulo para un correcto funcionamiento, se mantiene a la escucha, cuando escucha la orden de un patrón pregrabado y memorizado ejecuta la función action(), que como hemos mencionado anteriormente se ocupará del tratamiento específico para sonido.

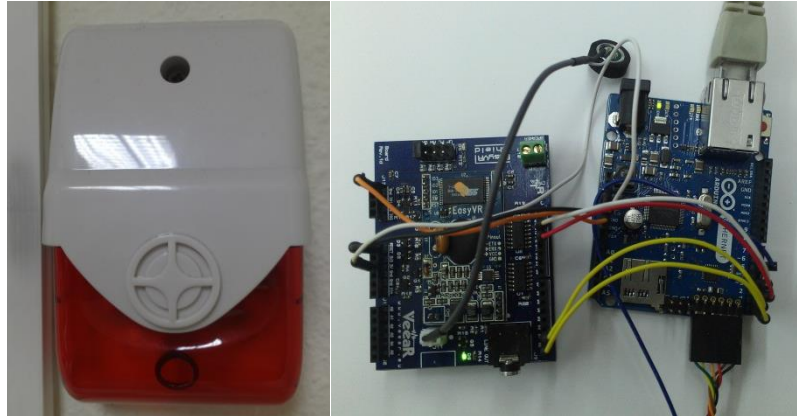
Este algoritmo se ejecutará sucesivamente por cada uno de los sonidos que se produzcan cerca del entorno del micrófono hasta el apagado del

microcontrolador.

### 3 Caso de Estudio: Desarrollo de un entorno inteligente para detección de alarmas.

El sistema se ha desarrollado en un laboratorio del grupo de investigación. En el laboratorio se tienen diferentes tipos de sensores como luminosidad, presencia por calor, sirenas y otros tipos de sensores todos conectados inalámbricamente mediante una red de sensores inalámbrica. Los sensores no se encuentran conectados al sistema en desarrollo puesto que se quiere poder integrar el sistema en los hogares sin necesidad de tener que gastar dinero en conectar hardware como los detectores de gas, gas o movimiento a la red de sensores. La conexión de los sensores y el sistema se realiza a través de las sirenas o timbres. El sistema detecta y clasifica los diferentes sonidos a partir de módulos de reconocimiento de sonido que interpretan los sonidos y ejecutan las acciones a partir de los agentes que se han embebido en los dispositivos. De este modo, se permite integrar el sistema en los hogares sin necesidad de un gasto elevado.

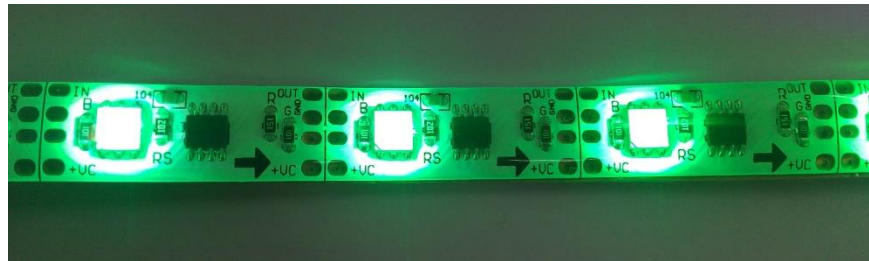
En la figura 4a) se puede ver la imagen de una de las sirenas instaladas en el laboratorio, en la figura 4b) se puede ver el EasyVR conectado a la placa arduino.



**Fig. 17.** a) Sirena de alarma b) EasyVR

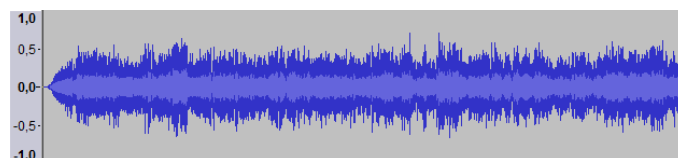
Durante las pruebas realizadas se tenían las siguientes alarmas: detección de humo, detección de gas, detección de robo, timbre, la activación de estas alarmas tenía consecuencias en las bandas luminosas.

En la habitación se tiene un dispositivo arduino según lo indicado en la sección 2.1. El agente se encuentra conectado a la plataforma PANGAEA que se encuentra instalada remotamente en un servidor. La banda luminosa se encuentran conectada al arduino. La figura 5 muestra la banda luminosa activada después de detectar una alarma.

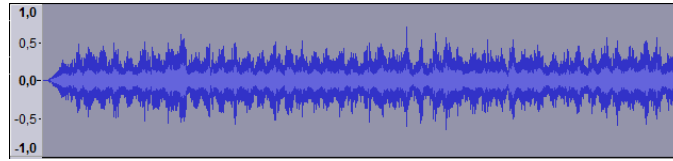


**Fig. 18.** Banda luminosa para advertir de la activación de una alarma

Finalmente para analizar el reconocimiento de los sonidos se procedió a grabar el sonido original de la alarma. El espectro de la alarma se puede ver en la figura 6a), en la figura 6b) se tomó el sonido de base y se le aplicó una distorsión denominada tremolo para modificar las frecuencias de los sonidos, al reproducir la nueva señal se podía apreciar la diferencia. Se comprobó que el sistema distinguía entre ambas señales.







**Fig. 19.** a)Sonido original b)sonido obtenido tras aplicar un filtro tremolo al sonido original.

## 4 Conclusiones

La arquitectura PANGAEA permite interconectar diferentes dispositivos facilitando así la integración de la información y la toma de decisiones a partir de la información proporcionada por los sensores. La arquitectura permite intercomunicar hardware de diferente tecnología para conseguir un procesamiento complejo que es inviable de realizar en un hardware con capacidad de procesamiento reducida como la plataforma arduino.

Una de las principales novedades del sistema ha sido la incorporación de agentes en dispositivos arduino permitiendo de este modo la incorporación sencilla de diferentes sensores y actuadores a la plataforma PANGAEA.

Arduino junto con el módulo de reconocimiento por voz permite reconocer diferentes sonidos a partir de los cuales se puede programar diferentes comportamientos. Esta característica es de relevancia ya que posibilita la incorporación al sistema de dispositivos como timbres, alarmas de humo, robo, gas mediante el reconocimiento de los sonidos. Este sistema reduce costes ya que se puede incorporar cualquier dispositivo que emita sonidos de alarma. Como resultados del reconocimiento de las alarmas se puede reproducir diferentes acciones como activación de bandas luminosas, envío de mensajes de alerta o activación de una cámara IP para realizar un seguimiento de la incidencia de modo remoto.

Una de las principales ventajas del sistema es el bajo coste de instalación ya que no es necesario tener ni un ordenador encendido y el hardware usado es de coste reducido. El coste más elevado está asociado a la conexión a internet que es necesaria para realizar la monitorización del entorno.

## Agradecimientos

Este trabajo ha sido soportado por el ministerio de ciencia e innovación en el proyecto iHAS (TIN2012-36586-C03-03).

## Referencias

- [1] G.T. Jayaputera, A. Zaslavsky, S.W. Loke, Enabling run-time composition and support for heterogeneous pervasive multi-agent systems, *Journal of Systems and Software*, 80 (2007) 2039-2062.
- [2] R. Want, A. Hopper, V. Falcao, J. Gibbons, The Active Badge Location System. *ACM Transactions on Information Systems*, 10(1) (1992) 91–102.



- [3] C.H. Lim, P. Anthony, L.C. Fan. Applying multi-agent system in a context aware. *Borneo Science* 24, (2009) 53-64.
- [4] A. Ahmed, J. Ali, A. Raza, G. Abbas, Wired Vs Wireless Deployment Support For Wireless Sensor Networks, *TENCON 2006. 2006 IEEE Region 10 Conference*, 2006, pp. 1-3.
- [5] M. Lee, H. Yoe, Comparative Analysis and Design of Wired and Wireless Integrated Networks for Wireless Sensor Networks, *Software Engineering Research, Management & Applications*, 2007. *SERA 2007. 5th ACIS International Conference on*, 2007, pp. 518-522.
- [6] J. Bajo, J.M. Corchado, Y. De Paz , J.F. De Paz, S. Rodríguez, Q. Martin, A. Abraham, *SHOMAS: Intelligent Guidance and Suggestions in Shopping Centres. Applied Soft Computing. Elsevier Science*, 9 (2) (2009) 851-862.
- [7] S.P. Fonseca, P. Martin, L. Griss, R. Letsinger, An Agent Mediated E-Commerce Environment for the Mobile Shopper, *Hewlett-Packard Laboratories, Technical Report, HPL-2001-157*, 2001.
- [8] R. Kowalczyk, M. Ulieru, R. Unland, Integrating Mobile and Intelligent Agents in Advanced e-Commerce: A Survey. In *Proceedings of the Agent Technologies, Infrastructures, Tools, and Applications for E-Services, NODE 2002, Lecture Notes in Computer Science*, Vol. 2592, Springer, Berlin, 2002, pp. 295-313.
- [9] S. Rodríguez, B. Pérez-Lancho, J.F. De Paz, J. Bajo and J.M. Corchado, *Ovamah: Multiagent-based Adaptive Virtual Organizations. . In Proceedings of the 12th International Conference on Information Fusion, Seattle, Washington, USA, 2009*, pp. 990-997
- [10] S.F. Barret, *Arduino microcontroller processing for everyone!. Morgan & Claypool Publishers*, 2010.
- [11] C. Carabelea, O. Boissier, Multi-Agent Platforms on Smart Devices : Dream or Reality? In *Proceedings of the SOC 2003*, 2003 pp. 126–129.
- [12] C. Fok, G. Roman, C. Lu. Agilla, A Mobile Agent Middleware for Self-Adaptive Wireless Sensor Networks. *ACM Transactions on Autonomous and Adaptive Systems (TAAS)* 4(3) (2009) 1-26.
- [13] G.G. Meyer, K. Främling, J. Holmström, Intelligent Products: A survey, *Computers in Industry*, 60 (3) (2009) 137-148.
- [14] E. Kazanavicius, V. Kazanavicius, L. Ostaseviciute, Agent-based framework for embedded systems development in smart environments. In *Proceedings of the 15th International Conference on Information and Software Technologies IT (2009)* 194-200.
- [15] R. Tynan, A. Ruzzelli, G.M.P. O'Hare, A Methodology for the Deployment of Multi-Agent Systems on Wireless Sensor Networks. *Mutiagent and Grid Systems*. (2006) 491-503.
- [16] A. Rogers, N.R. Jennings, D. Corkill, Agent Technologies for Sensor Networks. *IEEE Intelligent Systems* 24(2) (2009) 13-17.
- [17] F. Bellifemine, G. Cairea, A. Poggib, G. Rimassac *JADE: A software framework for developing multi-agent applications. Lessons learned. Information and Software Technology*, 50 (1-2) (2010) 10-21.
- [18] A Hardy, F Bouhafs, M Merabti. A Survey of Communication and Sensing for Energy Management of Appliances. *International journal of advanced engineering sciences and technologies* 3(2) (2010) 61-77.
- [19] Arduino Website <http://www.arduino.cc/>, last visited 21/09/2012.

- [20] Propeller Microcontroller Website <http://www.parallax.com/propeller/>, last visited 18/06/2013.
- [21] BeagleBoard Website <http://beagleboard.org/>, last visited 20/07/2012
- [22] C. Zato, G. Villarrubia, A. Sánchez, I. Barri, E. Rubión, A. Fernández, C. Rebate, J.A. Cabo, T. Álamos, J. Sanz, J. Seco, J., J. Bajo, J.M. Corchado, PANGEA - Platform for Automatic coNstruction of orGanizations of intElligent Agents. In Proceedings of the DCAI, Vol. 151, Springer, 2012, pp. 229-239
- [23] J. Ferber, O. Gutknecht, F. Michel, From Agents to Organizations: an Organizational View of Multi-Agent Systems, In Proceedings of the Agent-Oriented Software Engineering VI, LNCS Springer-Verlag. 2935, 2006, pp. 214–230.
- [24] Arduino Duemilanove Website <http://arduino.cc/es/Main/arduinoBoardDuemilanove/>, last visited 26/06/2013
- [25] Atmel168 <http://www.atmel.com/devices/atmega168.aspx>, last visited 20/06/2013
- [26] Integra Serial Control Protocol <https://sites.google.com/a/webarts.ca/toms-blog/Blog/new-blog-items/javaeiscp-integraserialcontrolprotocol/>, last visited 19/06/2013
- [27] Introduction to Pulse Width Modulation <http://www.barrgroup.com/Embedded-Systems/How-To/PWM-Pulse-Width-Modulation> last visited 23/09/2012
- [28] EasyVR Voice Recognition Module <https://www.sparkfun.com/products/10685>, last visited 21/06/2013
- [29] Universal asynchronous receiver/transmitter [http://en.wikipedia.org/wiki/Universal\\_asynchronous\\_receiver/transmitter](http://en.wikipedia.org/wiki/Universal_asynchronous_receiver/transmitter)